# Java 技术管理规范文档

# 一、命名规范

# **a、强制**

# 类名：使用大驼峰命名法（如UserService），禁止使用拼音或无意义缩写。

# 方法名：使用小驼峰命名法（如getUserInfo），首字母小写。

# 常量名：全部大写，单词间用下划线分隔（如MAX\_PAGE\_SIZE）。

# 包名：全部小写，采用域名倒序 + 模块名（如com.example.service）。

# 变量名：禁止使用单个字符（如a, b, c），必须语义清晰。

# **b、推荐**

# 接口名：推荐使用I前缀（如IUserDao）。

# 抽象类名：推荐使用Abstract前缀（如AbstractHandler）。

# 测试类名：使用Test后缀（如UserServiceTest）。

# 枚举类名：使用Enum后缀（如ErrorCodeEnum）。

# 集合变量名：使用复数形式（如userList, roleMap）。

# **c、允许**

# 临时变量：允许使用简短名称（如i, j作为循环计数器）。

# 内部类名：允许使用数字或简短描述（如UserHolder1）。

# 二、代码格式

# **a、强制**

# 方法分隔：类定义中方法之间必须空一行。

# 大括号位置：左大括号{不换行，右大括号}单独一行。

# 行长度限制：每行不超过 120 个字符，超过则换行。

# 参数分行：方法参数超过 3 个时必须分行书写。

# 注释语言：使用英文注释（特殊情况除外）。

# **b、推荐**

# 变量定义：方法内部变量集中在开头定义。

# 运算符空格：二元运算符两边必须有空格（如a + b）。

# 代码格式化：使用 IDE 自动格式化功能（如Ctrl+Alt+L）。

# 方法长度：单个方法不超过 50 行，超过则拆分。

# **c、允许**

# 注释风格：单行用//，多行用/\* \*/。

# 空行数量：允许连续不超过 2 个空行。

# 三、注释规范

# **a、强制**

# 类注释：必须包含 Javadoc，说明用途和注意事项。

# 方法注释：必须包含参数、返回值和异常说明。

# 公共成员注释：所有public和protected成员必须注释。

# 复杂逻辑注释：关键业务逻辑必须添加解释。

# **b、推荐**

# 私有方法注释：推荐添加功能说明。

# 常量注释：推荐说明用途。

# TODO 标记：使用// TODO:标记未完成功能。

# **c、允许**

# 废弃标记：使用@Deprecated注解标记过时代码。

# 抑制警告：使用@SuppressWarnings时必须说明原因。

# 四、依赖管理

# **a、强制**

# 构建工具：必须使用 Maven 或 Gradle 管理依赖。

# 版本明确：禁止使用latest或+作为依赖版本。

# 依赖检查：禁止引入重复依赖，使用dependency:tree检查。

# **b、推荐**

# 版本更新：定期使用插件检查过时依赖。

# 依赖统一：将公共依赖提取到父 POM 或 BOM 文件。

# **c、允许**

# 范围依赖：允许使用provided范围（如 Servlet API）。

# 快照版本：允许开发环境使用 SNAPSHOT 版本，但发布前必须更新。

# 五、异常处理

# **a、强制**

# 异常声明：方法抛出的异常必须在throws中声明。

# 异常处理：禁止捕获异常后不处理，至少记录日志。

# 自定义异常：自定义异常必须继承RuntimeException。

# **b、推荐**

# 统一处理：使用@ControllerAdvice统一处理异常。

# 异常信息：包含上下文信息（如用户 ID、操作时间）。

# **c、允许**

# 测试异常：允许在测试中使用@Test(expected = Exception.class)。

# 六、版本控制

# **a、强制**

# 版本工具：必须使用 Git 进行版本控制。

# 代码审查：提交前必须通过 Code Review。

# 提交规范：提交信息格式为[类型] 描述（如[FEATURE] 添加用户注册）。

# **b、推荐**

# 工作流程：使用 Git Flow 或 GitHub Flow 工作流程。

# 分支命名：功能分支使用feature/xxx格式。

# **c、允许**

# Rebase 操作：允许使用git rebase，但需谨慎。

# 忽略文件：使用.gitignore忽略不必要的文件。